Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 35 (2014) 397 – 406
Procedia Computer Science

18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - KES2014

Ensemble classifier for mining data streams

Ireneusz Czarnowski*, Piotr Jędrzejowicz

Gdynia Maritime University, Morska Str. 83, 81-225 Gdynia, Poland

Abstract

The problem addressed in this paper concerns mining data streams with concept drift. The goal of the paper is to propose and validate a new approach to mining data streams with concept-drift using the ensemble classifier constructed from the one-class base classifiers. It is assumed that base classifiers of the proposed ensemble are induced from incoming chunks of the data stream. Each chunk consists of prototypes and can be updated using instance selection technique when a new data have arrived. When a new data chunk is formed, ensemble model is also updated on the basis of weights assigned to each one-class classifier. The proposed approach is validated experimentally.

Peer-review under responsibility of KES International.

Keywords: data stream, one-clas classification, classifier ensemble

1. Introduction

The key objective of the machine learning is to design algorithms that are able to improve performance at some task through experience[1]. Such algorithms are called learners. Learning from examples is one of the most popular paradigms of the machine learning. Learning from examples is based on the existence of certain real-world *concepts* which might or might not be stable during the process of learning[1,2]. Traditional approach to learning from examples assumes that data under analysis have static distribution and the learning process remains unchanged during the time of the learner operation[4].

In many real-world situations, the environment in which a learner works is dynamic, i.e. the target concept and its statistical properties change over time and these changes are difficult to be predicted in advance. Such a property is typical in case of the, so called, *data streams*, where the class distribution of the stream data is imbalanced. Changes of data properties occurring in time are usually referred to as a data drift or a dynamic character of the data source[3,5]. Such changes are also known as a *concept drift*[6] or dataset shift[7], and learning in such case is referred to as *learning drift concept*[8] or *learning classifiers from the stream data*[9].

* Corresponding author. Tel.: +48 58 690 15 54.
E-mail address: irek@am.gdynia.pl.

Peer-review under responsibility of KES International.
doi:10.1016/j.procs.2014.08.120

Learning classifiers from data streams is a one of the recent challenges in data mining. Example of data streams, where concepts drift is natural, are sensor-produced streams, network logs, telephone call records, financial transactions and financial time series, traffic state, etc.[4]. The main feature of such streams is that data may be evolving over time. When standard machine learning algorithms used for processing data streams cannot efficiently handle changes in data, some specialized data stream mining methods are required. Their basic property is an ability to adopt to the concept drift automatically.

Processing streams of continuously incoming data implies a new computational requirement concerning a limited amount of memory and a short processing time, especially when data streams are large[3]. Data stream environments pose several further requirements on learning algorithms as compared with learning from static data. When dealing with data streams it is usually impossible to store all the data from streams and only a small part of the data can be stored and used for computations within a limited time span. In most cases, the arrival speed of the incoming instances from data streams enforce their processing in the real time[3]. An ideal learner for mining data streams should have the following properties: high accuracy, fast adaptation to change and low computation cost in both space and time dimensions[10].

To meet the above described requirements some summarization techniques for data streams mining have been proposed. Example of such techniques include the so called sampling or window models. Sampling is a most common technique used to decrease data size. Since, as a rule, data streams are not stationary and in times even unbounded, the sampling approach must be modified online. Such modifications could base on the analysis of data in each pass, that is after a new chunk of data has arrived, and removal of some instances from the training set with a given probability instead of periodically selecting them[11]. In[12] the idea has been extended to the weighted sampling method. Among other sampling-based approaches there are clustering of data streams and sampling within the sliding window model[13]. The last one bases on the assumption that analysis of the data stream is limited to the most recent instances, thus only limited number of instances is used to the learner training. In a simple approach sliding windows are of the fixed size and include most recent instances. With each new instance the oldest instance that does not fit window size is removed. When the window has a small size the classifier may react quickly to changes. Otherwise the classifier fails to adapt as rapidly as required to changes of data properties. Of course, decreasing the sliding window size may in turn lead to a loss of the mining algorithm accuracy.

Data summarization techniques can be also integrated with the drift detection techniques. The aim of the drift detection is to detect changes in the concept and inform the system that a learner should be updated or rebuild. Another approach dealing with a concept drift is the forgetting mechanism. The main idea is to select adequate data to be remembered. The approach assumes forgetting training instances at a constant rate and using only a window of the latest instances to train the classifier[14]. Alternatively, a selection of instances according to their class distribution can be used as the forgetting mechanism.

To deal with data streams, several online learning approaches, also termed incremental learning, are used for solving the classification tasks. In the online approach a classifier is also induced from the available training set. However, in addition, there is also some adaptation mechanism providing for a classifier evolution after the classification task has been initiated and started. In each round a class label of the incoming instance is predicted and afterwards information as to whether the prediction was correct or not, becomes available. Based on this information adaptation mechanism may decide to leave a classifier unchanged, or modify it, or induce a new one (see, for example, [15], [16]).

Among online approaches there are also ensemble methods, which build a new classifier when a new block of data arrives. Such a new classifier replaces the worst component in the current ensemble. Ensemble methods involve combinations of several models whose individual predictions are combined in some manner, using for example averaging or voting, to form a final prediction[10]. The family of ensemble algorithms for evolving data streams includes: Accuracy Weighted Ensemble, Adaptive Classifier Ensemble, Batch Weighted Ensemble, Streaming Ensemble Algorithm, Accuracy Diversified Ensemble and Weighted Majority Algorithm[17,18,14,19]. In[20] it has been proposed to differentiate incrementally build ensemble members using the so-called data chunks idea. In this approach the learner processes the incoming stream of data in data chunks. It is also observed that the size of these chunks is an important parameter and have influence on the learning quality.

A promising approach to online learning is to decompose a multi-class classification problem into a finite number of the one-class classification problems. The main feature of the one-class learning, in comparison with binary or multi-class classification, is that the learner tries to identify instances of a specific class amongst all instances, disposing of a training set containing only the instances of that class. This means that only one class of instances, called positive, is labelled in the training data. Thus one of the classes is well characterized by instances in the training data, while the other classes are not at all known or their presence in the training data is statistically insignificant [10]. One-class classification problem often applied to outlier detection or novelty detection, can be considered as a special case of binary classification. Thus, one-class classification algorithms aim to build classification models when the negative class is either absent, poorly sampled or not well defined [10]. The final goal is to predict whether a new instance falls into the same category as the positive examples or not. In [21] it has been underlined that one-class classification and novelty detection is well suitable for processing of data streams.

The main aim of the paper is to propose a new approach to the online learning based on weighted one-class classifier ensembles. In the paper, the set of one-class classification problems emerges from decomposition of the single multi-class classification problem. Thus multi-class classification problem is solved using an ensemble of single one-class classifiers, one per each target class. It is further assumed that such one-class base classifiers are induced from coming data chunks, consisting of prototypes. The prototypes are selected from the incoming instances using information as to whether the prediction of the incoming instances class has been correct or not. It is also assumed that data chunks are formed in accordance with requirements of the one-class learning method, and when a new instance arrives data chunks are updated using the instance selection technique. When a new data chunk is formed it enforces updating of the ensemble of classifiers.

The paper is organized as follows. Section 2 discusses the problem of learning from data streams. Section 3 explains main features of the proposed implementation of the ensemble model based on one-class classifiers. Section 4 provides details of the computational experiment results. In the final section conclusions and suggestions for future research are presented.

2. Online learning and problem definition

The online (or incremental) learning approaches are designed to sequentially learn a prediction model based on the feedback from answers to previous questions and possible additional side information [22]. In another words, during the classification, a class label of the incoming instance is predicted in each round, and afterwards information as to whether the prediction was correct or not is available. Based on this information the classifier can be updated to accommodate a new training instance [23].

Formally the algorithms are trying to solve online classification problem over a sequence of pairs $(x, c)_1, (x, c)_2, \dots,$ $(x, c)_t, \dots$, where x is an example's feature vector and c is its class label. The class label of each example can take any value from a finite set decision classes $C = \{c_{l:l=1,\dots,K}\}$, which has cardinality K. The sequence pairs can be provided to the learning algorithm as the training instances TR. At each time step t during training, the algorithm is used to find the best possible approximation f' of the unknown function f such that $f(x) = c$. Then f' can be used to find the class $c = f'(x)$ for any x such that $(x, c) \notin TR$.

After making a prediction the prediction model receives the feedback on the actual label. When the prediction was not correct the learning algorithm constructs the function for the next steps f'_{t+1} using f'_t and $(x, c)_t$.

3. Proposed approach

The proposed approach is based on the framework for data stream classification proposed by the authors in [24]. Based on this framework the online learning problem is solved focusing on data summarization, learning and classification. Data summarization process is responsible for extracting data chunks from the data stream instances. Afterwards data chunks are used to induce a classifier (learning phase). Finally, thus induced classifier is used to predict the class of instances whose class label are unknown. Online learning from data streams is based on the weighted one-class classifier ensemble. The following Sections discuss the proposed concepts of summarization, learning and classification.

3.1. Concept summarization

The classified data stream is given in a form of data chunks, with S_t denoting the tth data chunk and where S_t is a part of TR ($S_t \subset TR$ and $|S_t| < |TR|$). In the proposed approach instances included in S_t are seen as two subsets PS_t and US_t, independently for each one class $c_{l:l=1,\ldots,K} \in C$, where PS_t contains examples labeled as positive, with respect to the class c_l, and US_t contains unlabeled sample. Thus S_t is processed as a set of K subsets (data chunks), S_t^l consists of instances from PS_t^l and US_t^l (where $l = 1, \ldots, K$) and K one-class classifiers are induced from $PS_t^l \subset S_t^l$ and are used later on to predict the class of instances in S_{t+1}.

In the proposed approach data chunks consist of prototypes which are formed from sequence of incoming instances for which predictions were incorrect. It is also assumed that the size of chunks is not greater than the acceptable threshold. When a data chunk size is smaller than the threshold size, incoming instances are being added to this data chunk and are allocated to the corresponding S_t^i as positive instances and as negative instance to all other S_t^j, where $i = 1, \ldots, K$ and $i \neq l$, and the sum of sizes of these subsets is not grater then the threshold. When the size of the corresponding set S_t^l of instances labeled as positive, reaches the threshold, the chunk is updated.

In the paper, to update data chunks it is proposed to use the Edited Nearest Neighbor algorithm - as an instance selection technique, adopted to the considered one-class classification problem through applying the one-class k Nearest Neighbor method, called Nearest Neighbor Description (NN-d)[25]. Decision of accepting an instance as a member of the set of positive instances PS_t requires that its local density is greater than or equal to the local density of its nearest neighbor in the training set[25]. The pseudo-code of the proposed updating method, denoted ENN-d, is shown as Algorithm 1.

Input: S_t^l - the data chunk consisting of positive and negative instances ($S_t^l = PS_t^l \cup US_t^l$);
α - the value of the threshold;
k - the number of neighbors.
begin
  if $|S_t^l| > \alpha$ **then**
    Select randomly positive instance x_i from S_t^l;
    Select nearest neighbour x_i' for x_i in PS_t^l and set the distance d_1 between x_i and x_i';
    if $k = 1$ **then**
      Select nearest neighbour x_i^* for x_i' in PS_t^l and set the distance d_2 between x_i' and x_i^*;
    else
      Set the average distance d_2 to the k-nearest neighbours for x_i' in PS_t^l;
    end
    if $\frac{d_1}{d_2} > 1$ **then**
      $S_l^t := S_l^t \setminus \{x_i\}$;
    end
  end
  Return S_t^l; (Updated data chunk)
end

Algorithm 1: The Edited Nearest Neighbour algorithm based on the Nearest Neighbour Description (ENN-d)

3.2. Classifier ensemble

In this paper the pool of a simple base classifiers is represented by the matrix Φ consisting of $K \times \tau$ elements, i.e. K one-class classifiers, one per each target class, that represent history of τ earlier steps with respect to data chunks selection from the data stream:

$$\Phi = \begin{bmatrix} \phi_{t-\tau}^1 & \ldots & \phi_{t-1}^1 & \phi_t^1 \\ \vdots & \ddots & \vdots & \vdots \\ \phi_{t-\tau}^K & \ldots & \phi_{t-1}^K & \phi_t^K \end{bmatrix}, \quad (1)$$

where classifiers $\phi_t^l (l = 1, \ldots, K)$ are induced from set of positive instances $PS_t^l \subset S_t^l$ included in the sequentially arriving data chunks S_t^l. In the proposed approach, the ensemble consists of the fixed-size set of classifiers, depending on the value of τ (τ is a parameter set by the user). The ensemble is updated when a new data chunk arrives, i.e. when a new set S_t^j is available.

The updating process is associated with weights assigned to each of the base classifiers. The classifier's weight $w(\phi_t^l)$ is calculated from its accuracy and the length of time spent as a member of the ensemble. The value of weight increases in case the classifier has been taking the correct decisions. The value of classifier's weight is established, accordingly with the WAE approach[26], as follows:

$$w(\phi_t^l) = \frac{\Lambda(\phi_t^l)}{\sqrt{z}}, \tag{2}$$

where $\Lambda(\phi_t^l)$ denotes frequency of correct classification of classifier ϕ_t^l and z ($z \le \tau$) is a number of iteration for which ϕ_t^l stayed in the ensemble.

In the proposed approach ensemble model is updated by replacing the oldest set of one-class classifiers, induced on $S_{t-\tau}$, by a new set of classifiers, induced respectively on S_t. By default, it is assumed that each of the classifiers remains in the pool of ensemble no more than $\tau + 1$ steps. However, it has been proposed to promote good base one-class classifiers when their weights are above the average value of weights calculated for all classifiers in Φ belonging to the same class. According to this rule, when the weight assigned to oldest one-class classifier is higher than the average, then such one-class classifier remains active in the ensemble for the next stage. The pseudo-code of the procedure of constructing and updating the proposed weighted ensemble is shown as Algorithm 2.

```
Input: S_t - the data chunk consisting of positive and negative instances;
Φ - the matrix representing the ensemble of base classifiers;
W - the matrix of weights of classifiers from Φ;
τ - the column dimension size of Φ.
begin
    Decompose S_t into K subsets {PS_t^1 ∪ US_t^1}, ..., {PS_t^K ∪ US_t^K};
    for l = 1, ..., K and for all data chunks PS_t^l do
        Induce the classifier φ_t^{l*} from positive examples in PS_t^l using one of the standard approaches;
    end
    if Φ is empty or it's the column dimension is smaller than τ then
        ∀_{l=1,...,K} Φ ← Φ ∪ {φ_t^{l*}};
    else
        Calculate the average value of weights w̄ in W;
        for i = τ downton 1 do
            for l = 1, ..., K do
                if i = τ and w(φ_{t-i}^l) < w̄ then
                    φ_{t-τ}^l ← φ_{t-τ+1}^l;
                else
                    φ_{t-i}^l ← φ_{t-i+1}^l;
                end
            end
        end
        ∀_{l=1,...,K} Φ ← Φ ∪ {φ_t^{l*}};
    end
    Return Φ; (Updated ensemble model)
end
```

Algorithm 2: Constructing and updating weighted ensemble for the one-class classification

Finally, the prediction result produced by the proposed ensemble classifier is determined through the weighted majority vote[27]:

$$\Phi(x) = \arg\max_{l} \sum_{i=1}^{\tau+1} \sum_{l}^{K} w_i(\phi_i^l)(\phi_i^l(x) = c_l). \quad (3)$$

The procedure of data classification using the weighted ensemble for one-class classification is shown as Algorithm 3.

Input: $\mathcal{S}$ - the data stream of instances;
Φ - the matrix representing the ensemble of base classifiers;
W - the matrix of weights of classifiers from Φ.
begin
  for *all classifiers* $\phi_t^l \in \Phi$ **do**
    Apply ϕ_t^l on $x \in \mathcal{S}$;
  end
  Calculate prediction of ensemble model for according to (3);
  for *all classifier* $\phi_t^l \in \Phi$ **do**
    Calculate its weights $w(\phi_t^l)$ using (2) and them add $w(\phi_t^l)$ to W;
  end
  Return Φ, W;
end

Algorithm 3: Data classification using weighted ensemble for one-class classification

4. Computational experiment

This section contains the results of several computational experiments carried out with a view to evaluate the performance of the approach proposed in Section 3. The reported experiment aimed at answering the question whether the proposed approach can be useful tool to solve the data stream classification problems.

In this paper the proposed approach has been denoted as WECU (Weighted Ensemble with one-class Classification based on Updating of data chunk). The WECU has been also implemented in its version where the ensemble produces predictions using a simple majority voting without weights - such version of the algorithm is called WECUs.

The aim of the experiment has been also to compare WECU-based approaches with other algorithms for data stream mining:

- OLP-CNN, OLP-ENN and OLP-IB2 (OLP - Online Learning based on Prototypes), introduced by authors in[24], where data chunks have been updated using CNN[28], ENN[29] and IB2[30] algorithms, respectively. For these algorithms it has been decided to use a simple ensemble model in which ensembles are updated by removing the oldest classifier. The final output decision is produced through a simple majority voting to combine member decisions.
- Accuracy Weighted Ensemble (AWE), Hoeffding Option Tree (HOT) and Incrementally Optimized Very Fast Decision Tree (iOVFDT), which are implemented as extensions of the Massive Online Analysis package within WEKA environment[31].

In case of the WECU algorithm the one-class decision tree classifier called POSC4.5, introduced originally in[32], has been used as a base classifier. In case of the OLP family of algorithms the C4.5 algorithm[35] has been also applied to induce all of the base models for all ensemble classifiers. The value of the parameter denoting the number of base classifiers for each case has been set arbitrarily and was equal to 6 (i.e. $\tau = 5$). The ENN algorithm has been run with the number of neighbours equal to 3 (set arbitrarily). All implemented prototype selection algorithms (ENN, CNN and IB2) have been applied using the Euclidean metric.

All the above listed algorithms have been applied to solve the respective problems using several benchmark datasets obtained from the UCI Machine Learning [33] and IDA repositories[34]. Basic characteristics of these datasets are shown in Table 1. Table 1 includes also some results reported in the literature obtained using batch classifiers. Thresholds α for the size of data blocks (chunks) have been set up arbitrarily and values of the threshold are also shown in Table 1.

Generalization accuracy has been used as the performance criterion. An experiment plan has involved 30 repetitions of the proposed schema. The instances for the initial training set have been selected randomly from each considered dataset providing their number is not greater than a threshold. In each round, when a new instance arrives the algorithm predicts its class label. When the prediction is wrong an arrived instance is added to a data chunk or replace other one within the current data chunk according to the proposed approach.

Table 1. Datasets used in the experiment.

Dataset	Source of data	Number of instances	Number of attributes	Number of classes	Best reported results classification accuracy	Threshold (in % with respect to the original data set)
Heart	UCI	303	13	2	83.8% [36]	10%
Diabetes	UCI	768	8	2	80.12% [37]	5%
WBC	UCI	699	9	2	99.3% [37]	5%
Australian credit (ACredit)	UCI	690	15	2	92.1% [37]	9%
German credit (GCredit)	UCI	1000	20	2	80.3% [37]	10%
Sonar	UCI	208	60	2	97.1% [33]	10%
Satellite	UCI	6435	36	6	-	10%
Banana	IDA	5300	2	2	89.26% [36]	20%
Image	UCI	2310	18	2	80.3% [16]	20%
Thyroid	IDA	215	5	2	95.87% [36]	10%
Spambase	UCI	4610	57	2	82.37% [38]	20%
Twonorm	IDA	7400	20	2	97.6% [36]	20%

Table 2. Accuracy of the classification results (%).

Algorithm	WECU	WECUs	OLP-CNN	OLP-ENN	OLP-IB2	AWE	HOT	iOVFDT	C 4.5	SVM
Heart	84.14	81.10	78.14	80.40	81.50	78.01	81.40	81.70	77.80 [33]	81.50 [33]
Diabetes	79.62	76.75	73.20	72.82	71.22	72.5	80.42	77.40	73.00 [33]	77.00 [33]
WBC	72.54	71.40	70.10	71.21	72.40	72.81	72.67	71.04	94.70 [33]	97.20 [33]
ACredit	83.74	83.48	81.52	82.40	84.06	84.50	82.41	84.50	84.50 [33]	84.81 [33]
GCredit	75.40	73.04	70.06	71.84	71.30	73.50	72.87	75.21	70.50 [33]	72.50 [33]
Sonar	84.21	81.63	76.81	75.44	77.51	77.02	76.05	78.38	76.09 [33]	90.41 [33]
Satellite	79.14	80.75	80.45	78.25	76.14	82.40	83.40	81.54	-	85.00 [33]
Banana	88.10	86.45	84.10	85.72	88.10	87.40	86.77	89.21	88.55 [39]	90.04 [39]
Image	91.47	91.08	90.32	90.07	90.41	91.61	92.20	95.07	95.50 [39]	94.21 [40]
Thyroid	94.01	93.08	93.14	91.47	91.40	93.10	94.21	94.63	91.57 [41]	94.44 [42]
Spambase	78.50	78.50	77.61	78.17	77.80	75.40	77.40	80.20	92.34 [43]	92.5 [12]
Twonorm	97.70	96.81	95.10	96.40	95.72	76.80	96.06	97.00	97.14 [44]	97.13 [44]

Table 2 shows mean values of the classification accuracy of the classifiers obtained using the WECU approach. In Table 2 performances of the proposed approaches are also compared with performance of other online learning algorithms and two batch classifiers.

When two versions of the WECU algorithm are compared, as shown in Table 2, it can be observed that the best results have been obtained by the weighted ensemble approach. WECU outperforms its version without weights, where only a simple majority voting has been used. The observation holds true for all considered datasets. When the classification results of the family of WECU algorithms and the family of OLP algorithms are compared, a general observation is that the proposed one-class classification approach improves classification accuracy, at least with respect to the considered benchmark datasets.

The results shown in Table 2 further demonstrate that the WECU can be considered as superior to the other well-known methods including AWE, HOT and iOVFDT. This finding is supported by the fact that in three cases the proposed algorithm has been able to provide better generalization ability as compared to other methods.

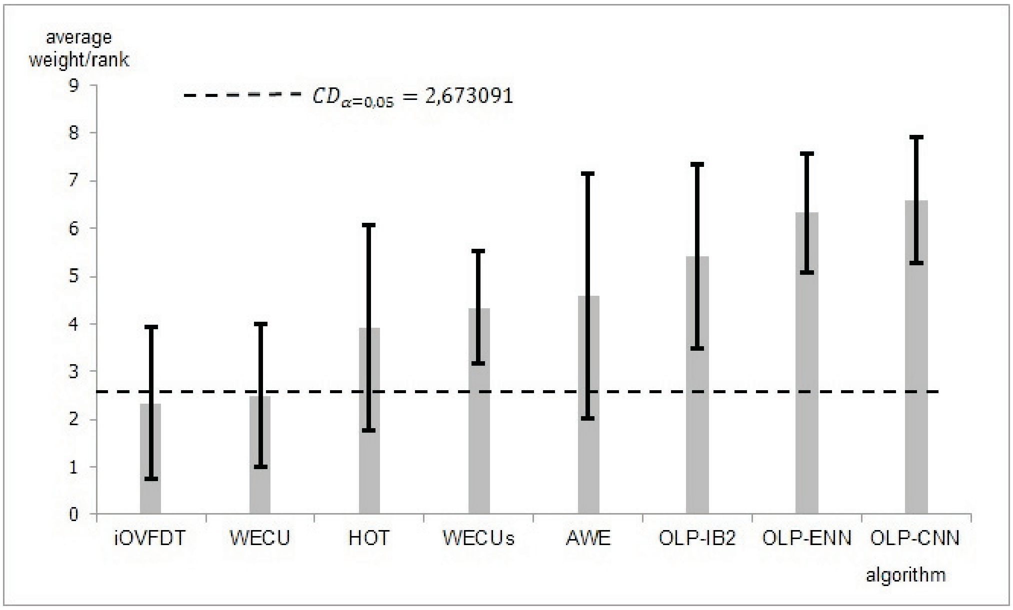

Fig. 1. The average Friedman test weights and the Bonferroni-Dunn's graphic corresponding to the obtained ranking.

Additionally, the Friedman non-parametric ranking test has been carried-out for comparison of the results. Results have been ranked and the ranking of the results has been computed assigning to the best of them rank 1 and rank 8 to the worst (the statistical analysis does not include results for SVM and C4.5). Fig. 1 depicts average weights and standard deviations for each compared algorithm obtained in Friedman test. Fig. 1 also shows the corresponding critical difference (CD) of Bonferroni-Dunn procedure. The horizontal cut line represents the threshold for the best performing algorithms. These bars which exceed the threshold are associated with algorithms displaying the worst performance with respect to the WECU. Thus it should be concluded that the WECU algorithm is beetter than HOT, WECUs, AWE and the OLP family of algorithms. From statistical point of view the WECU assures results not worse than the iOVFDT.

5. Conclusions

In this paper the ensemble classifier constructed from the one-class base classifiers is proposed for mining data streams with a concept-drift. Main feature of the proposed approach is inducing the base classifiers from data chunks formed from the incoming data stream. Data chunks are formed using algorithm adopted to suit requirements of the proposed one-class classifiers. The suggested ensemble classifier has been evaluated and compared with other approaches.

Basing on the reported computational experiment results it can be concluded that the proposed approach can be considered as a competitive approach to data stream classification. The approach extends the existing variety of methods applicable in field of the data streams mining. It is also a new addition to the family of the ensemble classifiers able to deal with the concept drift phenomenon. The paper confirms also that a prototype selection is a promising research direction when looking for effective stream mining tools.

Future research will also focus on studying influence of the size of both - ensemble model and data chunk on accuracy of the ensemble classifier.

References

1. Mitchell T. *Machine Learning*. McGraw-Hill, New York 1997.

2. Cichosz P. *Systemy uczące się*. Wydawnictwo Naukowo-Techniczne, Warszawa 2000 (in Polish).
3. Bifet A. *Adaptive learning and mining for data streams and frequent patterns*. PhD thesis, Universitat Politecnica de Catalunya; 2009.
4. Cazzolato MT, Ribeiro MX. A Statistical Decision Tree Algorithm Applied on Noisy Data Streams. Symposium on Knowledge Discovery, Mining and Learning - KDMiLe; 2013.
5. Sahel Z, Bouchachia A. 2007. Gabrys, B., Rogers, P.: Adaptive Mechanisms for Classification Problems with Drifting Data. In: Apolloni B. et al. (eds.) KES 2007, LNAI 4693, pp. 419-426. Springer-Verlag Berlin Heidelberg; 2007.
6. Widmer G, Kubat M. Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning* 23(1), 69-101; 1996.
7. Tsymbal A. *The Problem of Concept Drift: Definitions and Related work*. Tech. Rep. TCD-CS-2004-15, Department of Computer Science, Trinity College Dublin, Dublin, Ireland 2004.
8. Klinkenberg R. Learning Drifting Concepts: Example Selection vs. Example Weighting, Intelligent Data Analysis. *Incremental Learning Systems Capable of Dealing with Concept Drift* 8(3), 281-300; 2004.
9. Zhu X, Zhang P, Lin X, Shi Y. Active Learning from Data Streams. In: Proceedings of the Seventh IEEE International Conference on Data Mining, pp. 757-762; 2007.
10. Bifet A, Holmes G, Pfahhringer B, Kirkby R, Gavalda R. New Ensemble Methods For Evolving Data Streams. In KDD '09: Proceedings of the 15th ACM SIGKDD International conference on Knowledge Discovery and Data Mining, pp. 139-148, New York, NY, USA, ACM Press; 2009.
11. Vitter JS. Random sampling with a reservoir. *ACM Trans. Math. Software* 11(1), 37-57; 1985.
12. Chaudhuri S, Motwani R, Narasayya VR. On random sampling over joins. In: Delis A, Faloutsos C, Ghandeharizadeh S. (eds.), SIGMOD Conference, pp. 263-274. ACM Press; 1999.
13. Guha SG, Mishra N, Motwani R, O'Callaghan L. Clustering data streams. *FOCS*, pp. 359-366; 2000.
14. Kuncheva L. Classifier ensembles for changing environments. In: Roli F, Kittler J, Windeatt T. (eds.), Multiple Classifier Systems, LNSC 3077, pp. 1-15. Springer-Verlag Heidelberg; 2004.
15. Jędrzejowicz J, Jędrzejowicz P. Online Classifiers Based on Fuzzy C-means Clustering, Computational Collective Intelligence. Technologies and Applications. In: Badica C, Nguyen NT, Brezovan M, LNAI 8083, pp. 427 - 436, Springer, Berlin - Heidelberg; 2013.
16. Jędrzejowicz P, Jędrzejowicz J. A Family of the Online Distance-Based Classifiers. In: Nguyen NT, Attachoo B, Trawinski B, Somboonviwat K, (Eds.): Intelligent Information and Database Systems - 6th Asian Conference, ACIIDS 2014, Bangkok, Thailand, Proceedings, Part II. LNCS 8398, pp. 177-186, Springer, 2014.
17. Deckert M, Stefanowski J. Comparing Block Ensembles for Data Streams with Concept Drift. In Pechenizkiy M. et al. (eds.), New Trends in Database & Information Systems, AISC 185, pp. 69-78, Springer-Verlag Heidelberg; 2012.
18. Kuncheva L, Whitaker CJ. Measures of diversity in classifier ensembles. *Machine Learning* 51, 181-207; 2003.
19. Stefanowski J. Multiple and Hybrid Classifiers. In: Polkowski L. (ed.) *Formal Methods and Intelligent Techniques in Control, Decision Making. Multimedia and Robotics*, pp. 174-188, Warszawa; 2001.
20. Venkatesh G, Gehrke J, Ramakrishnan R. Mining Data Streams under Block Evolution. *SIGKDD Explorations* 3(2), 1-10; 2002.
21. Zhu X, Ding W, Yu PS. One-class learning and concept summarization for data streams. *Knowledge Information Systems* 28:523:553; 2011.
22. Shalev-Shwartz S. *Online learning: Theory, Algorithms, and Applications*, PhD thesis; 2007.
23. Jędrzejowicz J, Jędrzejowicz P. Cellular GEP-Induced Classifiers. In: Pan JS, Chen SM, Nguyen NT. (Eds.): ICCCI 2010, Part I, LNAI 6421, pp. 343-352. Springer-Verlag Berlin Heidelberg; 2010.
24. Czarnowski I, Jędrzejowicz P. Online Learning Based on Prototypes. In: Nguyen NT, Attachoo B, Trawinski B, Somboonviwat K. (Eds.): Intelligent Information and Database Systems - 6th Asian Conference, ACIIDS 2014, Bangkok, Thailand, April 7-9, 2014, LNAI 8398, pp. 187-196, Springer-Verlag Berlin Heidelberg; 2014.
25. Khan S, Madden MG. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 1-30, Cambridge University Press; 2014.
26. Wozniak M, Cal P, Cyganek B. The Influence of a Classifiers' Diversity on the Quality of Weighted Again Ensemble. In: Nguyen NT, Attachoo B, Trawinski B, Somboonviwat K. (Eds.): Intelligent Information and Database Systems - 6th Asian Conference, ACIIDS 2014, Bangkok, Thailand, 2014, LNAI 8398, pp. 90-99, Springer-Verlag Berlin Heidelberg; 2014.
27. Whang H, Fan W, Yu PS. Mining concept-drifting data streams using ensemble classifiers. In: Proceedings ACM SIGKDD, pp. 226-235; 2003.
28. Hart PE. The Condensed Nearest Neighbour Rule. *IEEE Transactions on Information Theory* 14, 515-516; 1968.
29. Wilson DR, Martinez TR. Reduction Techniques for Instance-based Learning Algorithm. *Machine Learning* 33(3), 257-286; 2000.
30. Aha DW, Kibler D, Albert MK, Instance-based Learning Algorithms. *Machine Learning* 6, 37-66; 1991.
31. Witten IH, Frank E. Data Mining: Practical Machine Learning Tools and Techniques (2nd edition), Morgan Kaufman, San Francisco, California, USA; 2005.
32. Comit'e De F, Denis F, Gilleron R, Letouzey F. Positive and Unlabeled Examples Help Learning, Algorithmic Learning Theory, Springer, pp. 219-230; 1999.
33. Asuncion A, Newman DJ. UCI Machine Learning Repository (http://www.ics.uci.edu/ mlearn/MLRepository.html). Irvine, CA: University of California, School of Information and Computer Science; 2007.
34. IDA Benchmark Repository, https://mldata.org/; 2014.
35. Quinlan JR. C4.5: Programs for Machine Learning. Morgan Kaufmann, SanMateo, CA; 1993.
36. Wang L , Hong-Bing J, Jin Y. Fuzzy Passive-Aggressive classification: A robust and efficient algorithm for online classification problems, *Information Sciences* 220:46-63; 2013.
37. Jędrzejowicz J, Jędrzejowicz P. Rotation Forrest with GEP-Induced Expression Trees, in: Shea JO', et al. (eds.) Agent and Multi Agent Systems: Technologies and Applications, LNAI, vol. 6682, pp. 495-503, Springer- Heidelberg; 2011.
38. Bertini JB, Zhao L, Lopes AA. An incremental learning algorithm based on the K-associated graph for non-stationary data classification, Information Sciences, 246:52-68; 2013.

39. Moreno-Torres JG, Saez JA, Herrera F. Study on the Impact of Partition-Induced Dataset Shift on k-fold Cross-Validation, *IEEE Transactions on Neural Networks and Learnig Systems* 23(8):1304-1312; 2012.
40. Sriwanna K, Puntumapon K, Waiyamai K. An Enhanced Class-Attribute Interdependence Maximization Discretization Algorithm, In: Zhu S. et al. (Eds.), ADMA 2012, LNAI 7713, pp.465-476, Springer-Verlag Berlin Heidelberg; 2012.
41. Sun Y. *Cost-sensitive boosting for classification of imbalanced data*, Ph.D. thesis, Waterloo University, Canada; 2007.
42. Kim HC, Ghahramani Z. Bayesian Gaussian process classification with the EM-EP algorithm, *IEEE Tranactions in Pattern Analysis and Machine Intelligence* 28, 1948-1959; 2006.
43. Sharma S, Arora A. Adaptive Approach for Spam Detection, *IJCSI International Journal of Computer Science Issues*, Vol. 10, Issue 4, No 1, 23-26; 2013.
44. Cai Y, Sun Y, Li J, Goodison S. Online Feature Selection Algorithm with Bayesian l_1 Regularization, In: Theerramunkong T, et al. (Eds.), Proceeding PAKDD '09 Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, LNAI 5476, pp.401-413 Springer-Verlag Berlin, Heidelberg; 2009.